Hybridization of Nonlinear and Mixed-Integer Linear Programming for Aircraft Separation With Trajectory Recovery

Jérémy Omer, Jean-Loup Farges

Hybridisation of Nonlinear and Mixed Integer Linear Programming for Aircraft Separation with Trajectory Recovery

Jérémy Omer, and Jean-Loup Farges

***Abstract*—The approach presented in this article aims at finding a solution to the problem of conflict-free motion planning for multiple aircraft on the same flight level with trajectory recovery. One contribution of this work is to develop three consistent models, from a continuous-time representation to a discrete-time linear approximation. Each of these models guarantees separation at all times as well as trajectory recovery, but they are not equally difficult to solve. A new hybrid algorithm is thus developed in order to use the optimal solution of a mixed integer linear program as a starting point when solving a nonlinear formulation of the problem. The significance of this process is that it always finds a solution when the linear model is feasible while still taking into account the nonlinear nature of the problem. A test bed containing numerous data sets is then generated from three virtual scenarios. A comparative analysis with three different initialisations of the nonlinear optimisation validates the efficiency of the hybrid method.**

***Index Terms*—Air traffic control, conflict resolution, nonlinear programming, mixed integer linear programming, optimal control.**

I. Introduction

AIR TRAFFIC Control (ATC) is the last stage of the current Air Traffic Management (ATM) system. It aims at monitoring the current air traffic in real-time. If necessary ATC is responsible for requesting changes in the trajectories of aircraft in order to maintain a reference horizontal separation of 5 NM or a vertical reference separation of 1000 ft. ATC is presently performed by human operators. The airspace is divided in geographical volumes and each volume is supervised by a pair of ATC Officers (ATCO). ATCOs are able to fulfill their task in the current network but the airspace is congested and many flights are delayed to keep air traffic density below manageable thresholds. The European research project on ATM, SESAR, points automation as a key feature for improvement of ATC [1].

A *conflict* between two aircraft is detected when both the horizontal and the vertical reference separation distances are not respected. Automated ATC systems usually aim at finding the trajectories of a set of aircraft such that a cost function is minimised and no pairwise conflict is detected in the next 10-15 minutes. In classical studies, trajectories are obtained by a sequence of manœuvres that each aircraft has to execute. These manœuvres may be classified into three categories: turns or *heading changes*, modifications of speed norm or *speed changes*, and vertical manœuvres.

The generic problem of collision avoidance is intrinsically difficult because it is nonconvex. Even with no constraint on motion dynamics, finding collision-free paths for rectangular objects was proved to be PSPACE-hard [2]. In addition, the mechanic and aerodynamic laws governing aircraft motion are very complex and finding the optimal trajectory of one aircraft is challenging in itself [3].

Conflict-free motion planning for aircraft has been the subject of many studies. The reader is referred to the remarkable review on automated ATC in [4] to get an extensive view on the problem. The state of the art given below illustrates some promising approaches.

Kirwan and Flynn [5], and Farley and Erzberger [6] remain in a framework very close to current operations. Efficient manœuvres are identified for pairwise conflicts and a heuristic search determines which action should be carried out. Such a process cannot guarantee that conflict-free trajectories are found, thus [5] plans to let ATCOs handle the situation in case of failure. Theoretical studies of the pairwise conflict with no bound on velocity [7] and with constant velocity [8] were carried out in the *optimal control* framework. Optimal trajectories were determined for this simple case and applied to the general case through a genetic algorithm [7] and a heuristic procedure [8].

Without prejudging what a good manœuvre should be, the problem may be simplified by allowing only one initial modification of speed vectors [9]–[11]. A Mixed Integer Linear Program (MILP) may then be solved to find conflict-free trajectories with speed modifications only, or with heading modifications only [9], or with both manœuvres [10]. [11] treats the 3D problem using mixed integer nonlinear programming. These approaches have to be included in a *receding horizon* procedure which iteratively solves the problem every t minutes in order to find the best speed vector at each time step. Unfortunately, each optimal solution is based on the assumption that only one speed modification will be carried out. Although, for instance, [10] takes into account the recovery of the initial trajectory in the objective function through an estimate of the additional distance travelled, these models do not explicitly include decision variables describing

Digital Object Identifier 10.1109/TITS.2013.2257758

J. Omer and J.-L. Farges are with the ONERA, the French Aerospace Lab, 2 av. Édouard Belin, 31000 Toulouse, France, e-mails: jeremy.omer@gmail.com, jean-loup.farges@onera.fr.

the manœuvres leading to recovery.

Alonso-Ayuso *et al.* [12] describe a MILP with multiple speed and altitude changes. They give an interesting analysis of tightening techniques implemented to solve their model more quickly. Richards and How [13] and Menon *et al.* [14] present complete models of the problem with multiple speed modifications. In [13], a MILP is solved after making the separation constraints linear while [14] keeps the original constraints and solves a Nonlinear Program (NLP). The strength of a MILP formulation is that its resolution converges to a globally optimal solution, but it only gives the optimum of an approximate model. On the other hand, nonlinear optimisation may take into account the real constraints but there is no guarantee that it leads to a feasible solution, and when it does, it only converges to a local optimum. Global optimisation techniques were developed in order to establish proofs of convergence when solving a nonconvex NLP. For instance, deterministic algorithms for global optimisation may involve some *divide and conquer* technique in order to explore the entire admissible space by sequentially solving sub-problems after partitioning the original space [15]. They usually require a great computational effort and may not be suited for this particular problem.

The main contributions of this article deal with modelling and algorithmic issues. As opposed to previous works involving a MILP formulation, a consistent evolution of the constraints, from their natural continuous-time representation to their discrete-time linear approximation, is presented. This rational approach allows for an analysis of the discretisation process and gives more insight on the differences between the different models. Additionally, constraints imposing trajectory recovery after the manœuvres are explicitly included. A hybrid algorithm is also developed in order to cope with the difficulties encountered when solving the nonlinear model. It takes advantage of the optimality of the MILP solutions in order to find a good starting point for the resolution of a NLP which considers the real geometry of the nonlinear constraints.

The basic problem formulation in continuous time is presented in Section II, Section III gives a NLP formulation to solve the problem numerically. An initialisation process through a MILP is given in Section IV, and improvements of this model, developed with a view to reducing computation time, are described in Section V. Finally, Section VI describes the benchmark on which the models are tested and reports the associated results.

II. Continuous time formulation

A. Working hypotheses

Several hypotheses are first stated in order to clarify the exact ATC problem that is tackled in this article.

Centralised planning: ATC is a centralised decision system. It is responsible for and has an authority on the whole controlled air space. The automated system presented here follows this conception of ATC. All the aircraft trajectories are planned at the same time, hence taking every possible interference between aircraft into account.

Planar motion: The current traffic is organised in horizontal layers, called *flight levels*, meeting the standard vertical separation. The traffic considered in this article is limited to aircraft flying on a high altitude, whose flight level is supposed to be stabilised. This portion of air traffic is also called *en-route traffic*. Manœuvres consisting in changing flight level to avoid a conflict are not popular, being both uneconomic, uncomfortable for passengers, and hard to monitor for ATCOs. This study then deals with planar motion planning of multiple aircraft flying on the same flight level.

4D contracts: With a view to improving the traffic predictability, several projects consider a trajectory-based ATM in which an aircraft would have to meet 4D waypoints along its flight [1], [16]. We define the *reference trajectory* of an aircraft as a planned trajectory that would satisfy every 4D waypoint. It is also assumed that each aircraft would follow its reference trajectory if there was no other aircraft to avoid. In order to allow for such 4D contracts, the problem solved here constrains the aircraft to recover their reference trajectories by the end of the conflict resolution time window.

Optimal trajectories: This problem aims at finding a cost minimising conflict-free motion planning. The cost of a solution is assumed to be proportional to the aircraft accelerations along their trajectories.

B. Kinodynamic planning

The motion planning problem deals with a set of aircraft $\mathcal{A}$ between which separation has to be maintained during a time interval $[0, T]$. The point-mass aircraft model described in [14] is commonly accepted to represent dynamical effects in civil aviation. Due to its complexity, there is a high risk that planning trajectories of several interfering aircraft with such a model would require too much computational effort. In the case of planar motion, a good approximation of the aircraft point mass model dynamics is derived by [8]. The state of an aircraft i at time t is represented by the vector $\mathbf{p}_i(t)$, whose Cartesian coordinates are $(p_{i,x}(t), p_{i,y}(t))$, and by its heading angle $\chi_i(t)$. The dynamics of the system to be controlled is given by: $\forall i \in \mathcal{A}, \forall t \in [0, T]$,

$$\begin{pmatrix} \dot{p}_{i,x}(t) \\ \dot{p}_{i,y}(t) \end{pmatrix} = V_i(t) \times \begin{pmatrix} \cos\chi_i(t) \\ \sin\chi_i(t) \end{pmatrix}$$
$$\dot{\chi}_i(t) = \omega_i(t),$$

where functions $t \mapsto \omega_i(t)$ and $t \mapsto V_i(t)$ are the control variables of the dynamical system. $V_i(t)$ is the horizontal velocity at t; $\omega_i(t)$ has the dimensions of an angular speed and is called *yaw rate*. In this model, [8] assumes that an autopilot is able to track the horizontal velocity and the yaw rate by adjusting the aircraft commands properly. This assumption is guaranteed by bounds on V and ω: $\forall i \in \mathcal{A}, \forall t \in [0, T]$,

$$\underline{V_i} \leq V_i(t) \leq \overline{V_i},$$
$$|\omega_i(t)| \leq \frac{\rho}{V_i},$$

where ρ is a constant issued from usual operational practice in terms of bank angles. As margins on speed are narrow in en-route traffic, the bound on yaw rate is approximated by the constant factor $\frac{\rho}{V_i^{ref}}$, where V_i^{ref} is the constant velocity of aircraft i on its reference trajectory.

In our approach, since polar coordinates of speed (V,χ) cannot be made linear, it is more relevant to favour a dynamical system expressed in Cartesian coordinates: $\forall i \in \mathcal{A}, \forall t \in [0,T]$,

$$\begin{pmatrix} \dot{p}_{i,x}(t) \\ \dot{p}_{i,y}(t) \\ \dot{v}_{i,x}(t) \\ \dot{v}_{i,y}(t) \end{pmatrix} = \begin{pmatrix} v_{i,x}(t) \\ v_{i,y}(t) \\ u_{i,x}(t) \\ u_{i,y}(t) \end{pmatrix}, \tag{1}$$

where $(v_{i,x}(t), v_{i,y}(t))$ and $(u_{i,x}(t), u_{i,y}(t))$ are the Cartesian coordinates of speed and acceleration at time t; the associated vectors are $\mathbf{v}_i(t)$ and $\mathbf{u}_i(t)$. The control variables are the two coordinates of acceleration. Bounds on the horizontal velocity are maintained as in [8] by, $\forall i \in \mathcal{A}, \forall t \in [0,T]$,

$$\underline{V_i}^2 \leq \|\mathbf{v}_i(t)\|^2 \leq \overline{V_i}^2, \tag{2}$$

where $\|\cdot\|$ is the Euclidean norm. The constraint on yaw rate is conserved by a bound on the norm of acceleration. By formulating the derivatives of $v_{i,x}^2(t)$ and $v_{i,y}^2(t)$ in the polar coordinates, it comes that

$$\|\mathbf{u}_i(t)\|^2 = \left(\dot{V}_i(t)\right)^2 + V_i^2(t)\omega_i^2(t)$$

Hence $|\omega| \leq \frac{\rho}{V}$ is guaranteed if, $\forall i \in \mathcal{A}, \forall t \in [0,T]$,

$$\|\mathbf{u}_i(t)\|^2 \leq \rho^2 \ \left(= \overline{U_i}^2\right) \tag{3}$$

C. Separation constraints

In this work, two aircraft i and j are said to be in *potential conflict* when the minimum distance between their reference trajectories is inferior to an arbitrary danger threshold (20 NM to 50 NM, depending on their relative direction). Let $\mathcal{C}$ be the set of pairs of aircraft in potential conflict. There is no conflict within the time window $[0,T]$ if and only if the following separation constraint is respected: $\forall (i,j) \in \mathcal{C}, \forall t \in [0,T]$,

$$\|\mathbf{p}_j(t) - \mathbf{p}_i(t)\|^2 \geq D^2, \tag{4}$$

where D is the horizontal reference separation distance.

A realistic formulation of the problem shall also account for the existence of prohibited areas, or *segregated areas*, in the airspace. Some portions of the airspace are indeed reserved, at least on a temporary basis, for military use. Rectangular motionless obstacles were already considered by [13]. The constraint in [13] is extended to any polygonal obstacle o delimited by a set of edges $\mathcal{E}_o$. Each polygon is then described by a set of affine inequalities $a_e x + b_e y + c_e \leq 0$, $e \in \mathcal{E}_o$. Let $\mathcal{O}$ be the set of obstacles; in order to avoid $o \in \mathcal{O}$ the position (p_x, p_y) of an aircraft has to verify at least one of the inequalities $a_e p_x + b_e p_y + c_e \geq 0$, $e \in \mathcal{E}_o$. Avoidance of all obstacles is thus guaranteed if and only if, $\forall o \in \mathcal{O}, \forall i \in \mathcal{A}, \forall t \in [0,T]$,

$$\max_{e \in \mathcal{E}_o} (a_e p_{i,x}(t) + b_e p_{i,y}(t) + c_e) \geq 0 \tag{5}$$

D. Complete problem

In addition to the constraints stated above,

- Initial position and speed of an aircraft i, $\mathbf{p}_i^{ini}$ and $\mathbf{v}_i^{ini}$, represent its present state and are supposed to be known.
- Each aircraft i has to recover its reference position and speed $\mathbf{p}_i^T$ and $\mathbf{v}_i^T$ by time T.
- An optimal motion planning minimises the norm of the control variables over $[0,T]$.

Tracking of the original plans is ensured by recovering reference positions and speeds at the end of the time interval. Most models found in the literature include a term ensuring the minimisation of flight time in their criterion [8], [11], [13], [14]. Due to trajectory recovery, such a term would be redundant here. Instead, we chose to minimise the norm of the control variables with a view to performing as smooth and small variations of speed and heading as possible. This criterion, or an equivalent one, is used in conjunction of other criteria by several authors [9], [13], [14]. Other objective functions based on fuel consumption or on deviation from reference trajectory could also be considered in a similar model.

The overall model may be formulated as a Bolza problem:

$$\mathcal{P} = \begin{cases} \min \quad z = \sum_{i \in \mathcal{A}} \left(\int_0^T \|\mathbf{u}_i(t)\| \,\mathrm{d}t \right) \\ \text{subject to constraints (1)-(5), and} \\ (\mathbf{p_i}(0), \mathbf{v}_i(0)) = (\mathbf{p}_i^{ini}, \mathbf{v}_i^{ini}), \ \forall i \in \mathcal{A} \\ (\mathbf{p}_i(T), \mathbf{v}_i(T)) = (\mathbf{p}_i^T, \mathbf{v}_i^T), \ \forall i \in \mathcal{A} \\ \mathbf{p}_i(t), \mathbf{v}_i(t), \mathbf{u}_i(t) \in \mathbb{R}^2, \ \forall t \in [0,T], \forall i \in \mathcal{A} \end{cases} \tag{6}$$

Partial solutions of $\mathcal{P}$ were found by [8], but they only exhibited portions of trajectories that are optimal for pairwise conflicts. No complete analytical solution of $\mathcal{P}$ was found yet.

III. Discrete time nonlinear model

In order to solve $\mathcal{P}$ numerically, the time window is discretised into a sequence of $K+1$ instants $0 = t_0 < t_1 < \ldots < t_K = T$. Converting the continuous time problem $\mathcal{P}$ into a discrete time problem is called *direct transcription* – see [17] for an overview of numerical resolution in optimal control. The model derived through direct transcription is a NLP. As far as modelling is concerned, two major difficulties arise during transcription. The differential equations (1) and the criterion of (6) have to be numerically integrated and the constraints have to be respected for all $t \in [0,T]$. In many studies on automated ATC involving numerical resolution, these issues are simplified by considering piecewise linear trajectories (e.g. [9], [10], [12]). These models consider that speed changes are instantaneous, which is only acceptable if manœuvres are initiated a long time before conflict [18]. After studying the error made by assuming speed changes are instantaneous, Paielli states that speed changes at constant acceleration were a much more acceptable option [18]. The numerical resolution of $\mathcal{P}$ was then simplified by assuming that acceleration is a piecewise constant function whose discontinuities correspond to the reception of control orders. If the sequence $\mathcal{T}$ is chosen so that new orders may only be issued at an instant $t \in t_0, \ldots, t_K$, acceleration is constant on each interval $[t_k, t_{k+1}[$.

In the rest of this paper, the following notations are used.

- $\mathcal{T} = \{t_k\}_{k\in\{0,\dots,K\}}$ and $\mathcal{T}^- = \{t_k\}_{k\in\{0,\dots,K-1\}}$
- $\mathcal{T}$ is partitioned according to a constant time step Δ : $\forall t_k \in \mathcal{T}^-, t_{k+1} - t_k = \Delta$.
- The value a function f takes at an instant $t_k \in \mathcal{T}$ is noted $f^k = f(t_k)$.

A. Numerical integration

Numerical integration of differential equations is commonly performed by a *collocation method* that approximates the solution by piecewise continuous functions such as polynomials of a chosen maximum order. In this particular case, it is assumed that acceleration is constant on each subinterval $[t_k, t_{k+1}[$. Differential equations (1) are then easy to solve exactly. Transcription of (1) leads to the exact expression:

$$\mathbf{p}_i^{k+1} = \mathbf{p}_i^k + \Delta \times \mathbf{v}_i^k + \frac{\Delta^2}{2}\mathbf{u}_i^k,\ \forall i \in \mathcal{A}, \forall t_k \in \mathcal{T} \tag{7a}$$

$$\mathbf{v}_i^{k+1} = \mathbf{v}_i^k + \Delta \times \mathbf{u}_i^k,\ \forall i \in \mathcal{A}, \forall t_k \in \mathcal{T}^- \tag{7b}$$

Acceleration being constant on each subinterval $[t_k, t_{k+1}[$, the criterion of $\mathcal{P}$ also has an exact expression in the discrete-time model:

$$z_{NL} = \sum_{i\in\mathcal{A}} \left(\sum_{k=0}^{K-1} \Delta \times \left\|\mathbf{u}_i^k\right\| \right)$$

B. State and control constraints

State constraints of the continuous-time problem $\mathcal{P}$ are given by upper bounds on velocity (2) and acceleration (3), separation constraints (4) and obstacle constraints (5). They have to be respected at any instant of the time horizon.

1) Bounds on speed and acceleration: In this model, acceleration of an aircraft i is piecewise-constant and speed is piecewise-linear. The two-variable function $(x, y) \to x^2+y^2$ is convex, so upper bounds on speed and acceleration are convex constraints. Hence it is sufficient to verify these constraints at each time step: $\forall i \in \mathcal{A}$,

$$\left\|\mathbf{v}_i^k\right\|^2 \leq \overline{V_i}^2,\ \forall t_k \in \mathcal{T} \text{ and } \left\|\mathbf{u}_i^k\right\|^2 \leq \overline{U_i}^2,\ \forall t_k \in \mathcal{T}^- \tag{8}$$

Lower bounds on speed lead to concave constraints but, as far as ATC is concerned, these bounds are set to keep speed in a small range around nominal speed. For instance, Paielli [18] states that speed decreases of more than 40 kt are often considered excessive. As this value remains a lot higher than the minimum speed induced by performance limitations, small violations of these constraints between two time steps may be accepted. Lower bound constraints are thus verified at each time step only:

$$\left\|\mathbf{v}_i^k\right\|^2 \geq \underline{V_i}^2,\ \forall i \in \mathcal{A}, \forall t_k \in \mathcal{T} \tag{9}$$

On the other hand, separation constraints (4) and obstacle constraints (5) do not tolerate any violation.

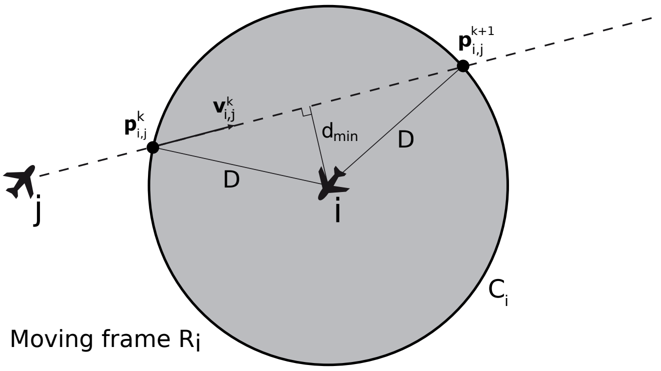


Fig. 1. Concavity of separation constraints and illustration of the worst case

2) Discrete verification of separation constraints: Let $(i, j) \in \mathcal{C}$ and $\mathbf{R}_i$ be the moving frame centred on i. Relative state variables of j in $\mathbf{R}_i$ are noted with double indices i, j, e.g. relative position is noted $\mathbf{p}_{i,j} = \mathbf{p}_j - \mathbf{p}_i$. A geometric representation, in frame $\mathbf{R}_i$, of the separation constraint for (i, j) is given in Figure 1. The constraint delimits a circular forbidden area $\mathbf{C}_i$ in the solution space for the state variable $\mathbf{p}_{i,j}$. Figure 1 illustrates that the minimum distance might be respected at t_k and t_{k+1} while not being respected inside the interval $[t_k, t_{k+1}]$.

It is important to get an idea of the error made when only checking separation constraints on a discrete time space. The order of magnitude of this error is obtained by studying a simple example where relative speed is assumed to be constant, i.e. $\mathbf{v}_{i,j}(t) = \mathbf{v}_{i,j}^k,\ \forall t \in [t_k, t_{k+1}]$. In that case, the worst situation happens when both $\left\|\mathbf{p}_{i,j}^k\right\| = D$ and $\left\|\mathbf{p}_{i,j}^{k+1}\right\| = D$ (see Figure 1). The minimum distance $\underline{d}_{i,j}^k$ between i and j is then reached at time t when the cartesian product of $\mathbf{p}_{i,j}(t)$ and $\mathbf{v}_{i,j}^k$ vanishes. Basic geometrical considerations lead to $t = t_k + \frac{\Delta}{2}$ and

$$(\underline{d}_{i,j}^k)^2 = D^2 - \left(\frac{\Delta}{2}\left\|\mathbf{v}_{i,j}^k\right\|\right)^2$$

Notice that this worst case often happens as it is the objective of the optimisation to find the shortest path. An extra margin D_m should then be added to D to ensure that i and j are always separated by the proper distance, with

$$(D + D_m)^2 = D^2 + \left(\frac{\Delta}{2}\left\|\mathbf{v}_{i,j}^k\right\|\right)^2$$

A numerical example with $D = 5$ NM, $\Delta = 1$ min and $\left\|\mathbf{v}_{i,j}^k\right\| = 500$ kt results with an additional margin $D_m = 4.71$ NM, which would be highly inefficient. As a consequence, it is important to find a constraint that guarantees separation at every instant.

3) Valid separation constraints: Due to the hypothesis of constant acceleration, the distance between aircraft i and j at time $t_k + \tau$, $\tau \in [0, \Delta]$, is:

$$d_{i,j}(t_k + \tau) = \left\|\mathbf{p}_{i,j}^k + \tau\mathbf{v}_{i,j}^k + \frac{\tau^2}{2}\mathbf{u}_{i,j}^k\right\|$$

In order to find $\tau_{i,j}^k$ so that the minimum distance $d_{i,j}^{min}$ is equal to $d_{i,j}(t_k + \tau_{i,j}^k)$, it is necessary to find the roots of the polynomial

$$\dot{d^2_{i,j}}(t_k+\tau) = 2\langle \mathbf{p}^k_{i,j} | \mathbf{v}^k_{i,j}\rangle + 2\left(\langle \mathbf{p}^k_{i,j} | \mathbf{u}^k_{i,j}\rangle + \langle \mathbf{v}^k_{i,j} | \mathbf{v}^k_{i,j}\rangle\right)\tau \\ +3\langle \mathbf{v}^k_{i,j} | \mathbf{u}^k_{i,j}\rangle\tau^2 + \langle \mathbf{u}^k_{i,j} | \mathbf{u}^k_{i,j}\rangle\tau^3,$$

where $\langle\cdot|\cdot\rangle$ is the Euclidean scalar product. Separation constraints thus involve the roots of a 3^{rd} degree polynomial. The number of distinct real roots of this polynomial depends on the sign of the discriminant. If included in the model, these constraints would have to be written as disjunctions of highly nonlinear expressions. Large scale problems with such constraints are usually too difficult to be solved efficiently.

Let us first consider the case where velocity is constant on a subinterval $[t_k, t_{k+1}]$, i.e. $\mathbf{u}_{i,j}(t_k+\tau) = 0$ for $\tau \in [0, \Delta[$. If velocity is constant, the only root of $\dot{d^2_{i,j}}(t_k+\tau)$ is easily obtained as

$$\tau^k_{i,j} = -\frac{\langle \mathbf{v}^k_{i,j} | \mathbf{p}^k_{i,j}\rangle}{\langle \mathbf{v}^k_{i,j} | \mathbf{v}^k_{i,j}\rangle}$$

Minimum distance between i and j on $[t_k, t_{k+1}]$ is reached at

$$\bar{t}^k_{i,j} = \begin{cases} t_k & \text{if } \tau^k_{i,j} \le 0 \\ t_{k+1} & \text{if } \tau^k_{i,j} \ge t_{k+1} \\ t_k + \tau^k_{i,j} & \text{otherwise} \end{cases} \tag{10}$$

The separation constraint between i and j on $[t_k, t_{k+1}]$ is then:

$$d^2_{i,j}\left(t_k + \max\left(0, \min(\Delta, \tau^k_{i,j})\right)\right) \ge D^2$$

The error made by assuming that speed is constant is now calculated. Let $\tilde{\mathbf{p}}_{i,j}(t)$ be an approximate position with constant speed and $\tilde{\mathbf{v}}^k_{i,j}$ the associated constant speed on $[t_k, t_{k+1}]$. $\tilde{\mathbf{p}}^k_{i,j}$ and $\tilde{\mathbf{v}}^k_{i,j}$ are chosen so that $\tilde{\mathbf{p}}_{i,j}(t_k) = \mathbf{p}^k_{i,j}$ and $\tilde{\mathbf{p}}_{i,j}(t_{k+1}) = \mathbf{p}^{k+1}_{i,j}$, *i.e.*:

$$\tilde{\mathbf{v}}^k_{i,j} = \mathbf{v}^k_{i,j} + \frac{\Delta}{2}\mathbf{u}^k_{i,j}$$

The error made when approaching $\mathbf{p}_{i,j}$ with $\tilde{\mathbf{p}}_{i,j}$ on $[t_k, t_{k+1}]$ is given by

$$\|\mathbf{p}_{i,j}(t_k+\tau) - \tilde{\mathbf{p}}_{i,j}(t_k+\tau)\| = \frac{1}{2}\tau(\Delta-\tau)\left\|\mathbf{u}^k_{i,j}\right\|,$$

which is maximal for $\tau = \Delta/2$. As a consequence, separation of all pairs of aircraft in potential conflict is guaranteed by: $\forall (i,j) \in \mathcal{C}, \forall t_k \in \mathcal{T}^-$,

$$\left\|\tilde{\mathbf{p}}_{i,j}(\bar{t}^k_{i,j})\right\|^2 \ge \left(D + \frac{\Delta^2}{8}\left\|\mathbf{u}^k_{i,j}\right\|\right)^2, \tag{11}$$

where $\bar{t}^k_{i,j}$ is calculated as in (10).

Typical values of $\overline{U}$ result in an error term $\frac{\Delta}{8}\left\|\mathbf{u}^k_{i,j}\right\| \le 1$ NM. Directly adding the error term in the constraint leads to an interesting behaviour in the overall traffic: aircraft mostly modify their speed before and after the potential conflicts so that the error term has no impact when close to the conflict. Conflict resolution is mostly done in this manner today. ATCOs send orders several minutes before a detected conflict, and clearance to recover initial trajectories is only sent once the conflict is sure to be solved.

4) Obstacle constraints: Polygonal constraints are problematic for aircraft i when the closest edge e for which $a_e p_{i,x} + b_e p_{i,y} + c_e \ge 0$ changes. Aircraft could be outside the obstacle at time t_k and t_{k+1} while crossing it in the meantime. A conservative option is chosen to avoid such a situation. For all aircraft i and all subintervals $[t_k, t_{k+1}]$, it is required that at least one edge stays with its constraint verified for the whole time interval. Similar calculations as above lead to constraints: $\forall i \in \mathcal{A}, \forall t_k \in \mathcal{T}$,

$$\max_{e\in\mathcal{E}_o}(\min(\quad a_e p^k_{i,x} + b_e p^k_{i,y} + c_e - m_e, \\ a_e p^{k+1}_{i,x} + b_e p^{k+1}_{i,y} + c_e - m_e)) \ge 0 \tag{12}$$

with m_e an additional margin taking into account non-constant speed. m_e depends on the angle between obstacle edges and is less than 1 NM for typical values of Δ, $\overline{V}$ and $\overline{U}$.

C. Summary of the NLP

The complete nonlinear model $\mathcal{P}_{NL}$ is summarised below:

$$\mathcal{P}_{NL} = \begin{cases} \min \quad z_{NL} = \sum_{i\in\mathcal{A}}\left(\sum_{k=0}^{K-1} \Delta\left\|\mathbf{u}^k_i\right\|\right) \\ \text{subject to constraints (7)-(12), and} \\ \forall i \in \mathcal{A}: \\ \quad \left(\mathbf{p_i}^0, \mathbf{v}^0_i, \mathbf{p}^K_i, \mathbf{v}^K_i\right) = \left(\mathbf{p}^{ini}_i, \mathbf{v}^{ini}_i, \mathbf{p}^T_i, \mathbf{v}^T_i\right), \\ \quad \mathbf{p}^k_i, \mathbf{v}^k_i \in \mathbb{R}^2, \forall i \in \mathcal{A}, \forall t_k \in \mathcal{T} \\ \quad \mathbf{u}^k_i \in \mathbb{R}^2, \forall i \in \mathcal{A}, \forall t_k \in \mathcal{T}^- \end{cases}$$

It is well known that the solution of a nonconvex NLP depends on the initial point from which it is started. Several global optimisation techniques rely on a stochastic generation of multiple initial points before running the NLP solver on each point [19]. Although it cannot prove the optimality of a given solution, stochastic global optimisation offers an asymptotic convergence guarantee in some probabilistic sense. These approaches thus constitute a faster alternative to deterministic global optimisation. On the other hand, such a *multi-start* technique still requires that the NLP is solved several times to perform well, which may already be far too costly for a large conflict-free planning of multiple aircraft trajectories. The option that was preferred in this article focused on generating a unique good starting point. $\mathcal{P}_{NL}$ is then approximated by a MILP for which a global optimum may be found in reasonable time for most operational cases.

IV. A Mixed Integer Linear formulation

A Linear Program (LP) is an optimisation problem formulated as the minimisation of a linear criterion subject to a set of linear constraints on continuous variables. LP is a very popular framework in optimisation because some algorithms, such as the Dantzig simplex [20] or the interior point algorithm [21], proved to be very efficient. Several works on conflict-free trajectories planning proposed a linear model for the automation of ATC [9], [13], [22], [23], for the motion of virtual humans [24], or for the general case of multiple robots [25].

Modelling this problem as a LP is not natural, due to the nonconvex separation constraints. The goal may be achieved

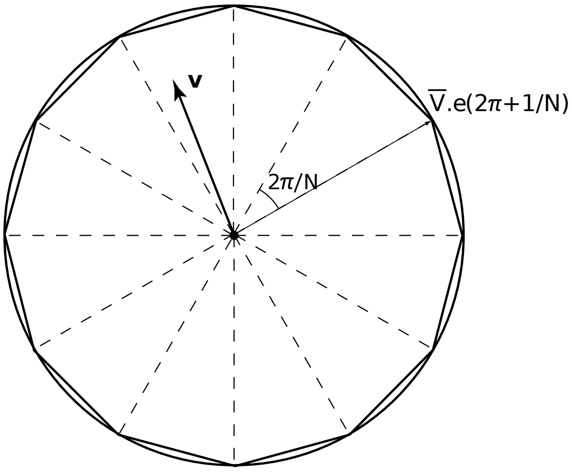


Fig. 2. Approximation of upper bounds with chords

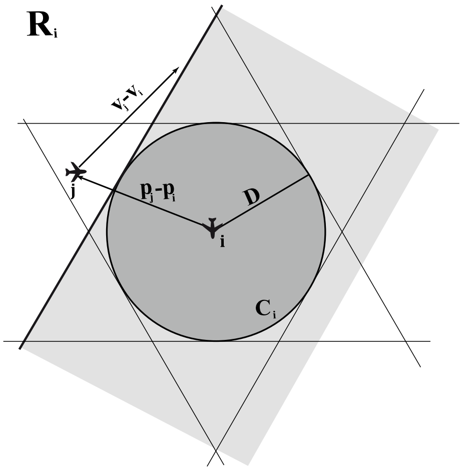


Fig. 3. Approximation of the separation circle with tangents

by making strong assumptions which are not consistent with the search of a solution close to the global optimum of $\mathcal{P}_{NL}$. The model is very close to a LP when deviations are plainly forbidden, *i.e.* $\omega = 0$ [9], when a set of decisions is done before resolution, as in [22], or under rules of reciprocal collision avoidance [24]. As the approached model aims at finding an initial solution for the NLP resolution, the same assumptions must be done in both formulations. Discussions below show that $\mathcal{P}_{NL}$ may be approximated with a linear criterion and linear constraints, but separation constraints involve continuous and binary variables, hence leading to a MILP.

A. Convex constraints on speed and acceleration

Upper bounds on acceleration and velocity are convex but they are quadratic. These constraints may be interpreted geometrically as circles inside of which speed and acceleration vectors have to lie. As in [22] and [13], an upper bound is approximated by N chords equally spread on the circle (cf. Figure 2).

Property 1: The relative error made by approximating a circle with N equally spread chords is $1-\cos\left(\frac{\pi}{N}\right)$.

Proof: Maximum error is reached on the middle of each chord. Basic geometry considerations lead to the result. ■

Let $\theta \in [0,2\pi]$: the vector $\mathbf{e}(\theta)$ is defined as $\mathbf{e}(\theta) = (\cos\theta, \sin\theta)$. For any vector $\mathbf{v}$, the set of equations

$$\langle \mathbf{v} | \mathbf{e}\left(\tfrac{(2n-1)\pi}{N}\right)\rangle \leq \overline{V}\cos\left(\tfrac{\pi}{N}\right), \quad n = 1,..,N$$

is a conservative linear approximation of the constraint $\|\mathbf{v}\| \leq \overline{V}$ with a maximum relative error $1-\cos\left(\frac{\pi}{N}\right)$. Although it might result in a great number of additional constraints, they only involve real variables, which are easy to handle when compared to binary variables.

This formulation does not allow to compute the nonlinear criterion, but a set of intermediate variables $\overline{u}_i^k$ might be used to get a good approximation of $\left\|\mathbf{u}_i^k\right\|$. The following approximate constraints on acceleration are then considered:

$$\begin{array}{c} \langle \mathbf{u}_i^k | \mathbf{e}\left(\tfrac{(2n-1)\pi}{N}\right)\rangle \leq \overline{u}_i^k \cos\left(\tfrac{\pi}{N}\right), \quad \forall i,k,n \\ 0 \leq \overline{u}_i^k \leq \overline{U}_i, \quad \forall i,k \end{array} \tag{13}$$

Instead of being directly bounded by $\overline{U}_i$, the norm of acceleration is bounded by $\overline{u}_i^k$, which is bounded by $\overline{U}_i$. If $\overline{u}_i^k$ is minimised, it gets as close as possible to the norm of acceleration. It is thus equivalent to minimise $\left\|\mathbf{u}_i^k\right\|$ and to minimise $\overline{u}_i^k$. This leads to a linear form of the criterion:

$$z_{MIL} = \sum_{i\in\mathcal{A}} \left(\sum_{k=0}^{K-1} \Delta \times \overline{u}_i^k \right)$$

Regarding velocity, it is convenient to save a great number of constraints by considering that not every direction is efficient or even reachable for short term conflict avoidance. Denote ψ_i^k the angle between $\mathbf{v}_i^k$ and the speed vector on reference trajectory at time t_k. It was empirically observed during tests on a reduced version of the benchmark described in Section VI that ψ_i^k remained in the range $[-\frac{\pi}{4}, \frac{\pi}{4}]$. Upper bounds on velocity are then respected if $2Q+1$ constraints are set in the range $[-\frac{\pi}{4}, \frac{\pi}{4}]$ around the reference speed. $\forall i \in \mathcal{A}, t_k \in \mathcal{T}, q = 0,..,2Q$:

$$\langle \mathbf{v}_i^k | \mathbf{e}\left(\tfrac{(q-Q)\pi}{4Q} + \psi_{i,k}^{\mathrm{ref}}\right)\rangle \leq \overline{V}_i \cos\left(\tfrac{\pi}{8Q}\right), \tag{14}$$

where $\psi_{i,k}^{\mathrm{ref}}$ is the direction of reference speed on $[t_k, t_{k+1}]$.

B. Nonconvex constraints

Approximating $\mathcal{P}_{NL}$ with a linear model actually consists in forming a disjunction of convex problems. Let us focus on a potential conflict between two aircraft i and j. Once represented in the moving frame $\mathbf{R}_i$, the separation constraint excludes the separation circle $\mathbf{C}_i$ (Figure 1). As the half-planes delimited by the straight lines tangent to $\mathbf{C}_i$ are the biggest convex sets excluding $\mathbf{C}_i$, it is adequate to approximate separation constraints by disjunctions of such half-planes.

The difficulty due to the quadratic evolution of position with respect to time is handled as for the nonlinear case. Conflicts are solved for aircraft with a constant speed $\tilde{\mathbf{v}}_{i,j}^k = \mathbf{v}_{i,j}^k + \frac{\Delta}{2}\mathbf{u}_{i,j}^k$ and the error made is compensated by adding a term to separation distance. The modified separation distance is:

$$D_{i,j}^k = D + \frac{\Delta^2}{8}(\overline{u}_i^k + \overline{u}_j^k)$$

If speed is constant on $[t_k, t_{k+1}]$, the motion of aircraft j in $\mathbf{R}_i$ is a straight line. Separation is thus guaranteed on $[t_k, t_{k+1}]$ if j is on the separating side of the same tangent at both t_k and t_{k+1}, as is represented in Figure 3. The constraint is conservative because it may force an aircraft to move away from reference trajectory although separation is already performed. This is however essential to maintain the proper separation at every moment. Eventually the separation constraint is approximated by a set $\mathcal{S}$ of tangents above one of which j has to be at both t_k and t_{k+1} to avoid conflict during $[t_k, t_{k+1}]$. Such disjunction is modelled with linear constraints by using binary variables. The optimisation decides which tangent maintains separation between i and j on $[t_k, t_{k+1}]$ by setting the values of binary variables $\delta^k_{i,j,s}$, $s \in \mathcal{S}$, such that $\delta^k_{i,j,s} = 1$ if j is on the side of the tangent s that does not include $\mathbf{C}_i$ during $[t_k, t_{k+1}]$ and $\delta^k_{ij,s} = 0$ otherwise. Let θ_s, $s \in \mathcal{S}$, be such that s is tangent to $\mathbf{C}_i$ at the point $D \times \mathbf{e}(\theta_s)$. Separation on $[t_k, t_{k+1}]$ is formulated as: $\forall t_k \in \mathcal{T}^-, \forall (i,j) \in \mathcal{C}$,

$$\langle \mathbf{p}^k_{i,j} | \mathbf{e}(\theta_s) \rangle \geq D^k_{i,j} - M(1 - \delta^k_{i,j,s}), \quad \forall s \in \mathcal{S} \tag{15}$$

$$\langle \mathbf{p}^{k+1}_{i,j} | \mathbf{e}(\theta_s) \rangle \geq D^k_{i,j} - M(1 - \delta^k_{i,j,s}), \quad \forall s \in \mathcal{S} \tag{16}$$

$$\textstyle\sum_{s \in S} \delta^k_{i,j,s} = 1 \tag{17}$$

$$\delta^k_{i,j,s} \in \{0,1\}, \forall s \in \mathcal{S} \tag{18}$$

The positive constant M is chosen big enough to ensure that constraints (15)-(16) are redundant for any relative position if $\delta^k_{i,j,s} = 0$. Constraints (17) make sure that at least one tangent is separating i and j on $[t_k, t_{k+1}]$. This technique is commonly used to formulate disjunctive constraints in a MILP. It is usually referred to as the *big-M* formulation.

Following a similar reasoning, avoidance of segregated areas $o \in \mathcal{O}$ is modelled with linear constraints involving binary variables $\delta^k_{i,e}$, $e \in \mathcal{E}_o$, such that aircraft i is on the side of edge e that does not include o during $[t_k, t_{k+1}]$ if $\delta^k_{i,e} = 1$, *i.e.*: $\forall i \in \mathcal{A}, \forall t_k \in \mathcal{T}^-, \forall o \in \mathcal{O}$,

$$a_e p^k_{i,x} + b_e p^k_{i,y} + c_e + m_e \geq M(\delta^k_{i,e} - 1), \, \forall e \in \mathcal{E}_o, \tag{19}$$

$$a_e p^{k+1}_{i,x} + b_e p^{k+1}_{i,y} + c_e + m_e \geq M(\delta^k_{i,e} - 1), \, \forall e \in \mathcal{E}_o \tag{20}$$

$$\sum_{e \in \mathcal{E}_o} \delta^k_{i,e} = 1, \tag{21}$$

The lower bound on speed is also ensured by a disjunctive constraint. If the quadratic constraint $\|\mathbf{v}\|^2 \geq \underline{V}^2$ is represented by a circle outside of which $\mathbf{v}$ has to lie, a linear approximation is obtained by considering a set of tangents instead of the chords involved in the upper bound constraints. The lower bound is thus satisfied if $\mathbf{v}$ satisfies at least one of the constraints associated with the tangents. This is formulated through constraints similar to (14) by adding a set of binary variables $\epsilon^k_{i,q}$: $\forall i \in \mathcal{A}, \forall t_k \in \mathcal{T}, \forall q \in \{0, \dots, 2Q\}$,

$$\langle \mathbf{v}^k_i | \mathbf{e}\left(\frac{(q-Q)\pi}{4Q} + \psi^{\text{ref}}_{i,k}\right) \rangle \geq \underline{V}_i - M'(1 - \epsilon^k_{i,q}) \tag{22}$$

$$\sum_{q \in \{0,\dots,2Q\}} \epsilon^k_{i,q} = 1 \tag{23}$$

$$\epsilon^k_{i,q} \in \{0,1\} \tag{24}$$

C. Summary of the MILP

The complete MILP $\mathcal{P}_{MIL}$ is given here:

$$\mathcal{P}_{MIL} = \begin{cases} \min. \quad z_{MIL} = \sum_{i \in \mathcal{A}} \left(\sum_{k=0}^{K-1} \Delta \times \overline{u}^k_i \right) \\ \text{subject to constraints (7), (13)-(24), and} \\ \forall i \in \mathcal{A}: \\ \left(\mathbf{p_i}^0, \mathbf{v}^0_i, \mathbf{p}^K_i, \mathbf{v}^K_i\right) = \left(\mathbf{p}^{ini}_i, \mathbf{v}^{ini}_i, \mathbf{p}^T_i, \mathbf{v}^T_i\right) \\ \mathbf{p}^k_i, \mathbf{v}^k_i \in \mathbb{R}^2, \, \forall i \in \mathcal{A}, \forall t_k \in \mathcal{T}, \\ \mathbf{u}^k_i \in \mathbb{R}^2, \overline{u}^k_i \in \mathbb{R}^+, \, \forall i \in \mathcal{A}, \forall t_k \in \mathcal{T}^-, \\ \delta^k_{i,j,s} \in \{0,1\}, \, \forall (i,j) \in \mathcal{C}, \forall t_k \in \mathcal{T}^-, \forall s \in \mathcal{S}, \\ \delta^k_{i,e} \in \{0,1\}, \, \forall i \in \mathcal{A}, \forall t_k \in \mathcal{T}^-, \forall e, \\ \epsilon^k_{i,q} \in \{0,1\}, \, \forall i \in \mathcal{A}, \forall t_k \in \mathcal{T}, \forall q \end{cases}$$

V. Computational analysis and improvements

$\mathcal{P}_{MIL}$ is a disjunction of a large number of LPs. By exploring the tree built from the possible decisions with an efficient algorithm such as a *branch and bound* algorithm, first described by [26], it is possible to find the global optimum of the overall problem. At each node of the tree, a *continuous relaxation* of $\mathcal{P}_{MIL}$ is solved by fixing a set of binary variables to chosen values and by replacing constraints $\delta \in \{0,1\}$ with $\delta \in [0,1]$ for the other variables. In this framework, computation time depends mostly on three criteria:

1) the size of the continuous relaxation, given by its number of constraints and variables;
2) the number of binary variables;
3) the tightness of continuous relaxations.

The first two aspects are quantified below. In the following property, $|\mathcal{S}et|$ refers to the cardinal of $\mathcal{S}et$, and the total number of edges is noted E, *i.e.* $E = \sum_{o \in \mathcal{O}} |\mathcal{E}_o|$.

Property 2: $\mathcal{P}_{MIL}$ is composed of

- $7\,|\mathcal{A}| \times K$ continuous variables for position, speed and acceleration,
- $K \times |\mathcal{S}| \times |\mathcal{C}|$ binary variables for separation and $|\mathcal{A}| \times K \times |\mathcal{O}|$ binary variables for obstacles avoidance,
- $|\mathcal{A}| \times (8 + K \times (4 + N + 4Q))$ motion constraints,
- $K \times |\mathcal{C}| \times (2\,|\mathcal{S}| + 2) + K \times |\mathcal{A}| \times (E + |\mathcal{O}|)$ separation and segregated area constraints.

While $|\mathcal{A}|$, $|\mathcal{C}|$, $|\mathcal{O}|$ and E are problem data, K, $|\mathcal{S}|$, N and Q are parameters of the model. As the model has to remain accurate, the problem's size may not be decreased below an acceptable threshold. The remaining option is to make continuous relaxations much tighter by calculating good values for big-M constants in (15)-(16) and by adding valid inequalities.

A. Tightening $\mathcal{P}_{MIL}$

The choice of a good value for big-M constants in (15)-(16) depends on the pair of aircraft, the instant and the tangent considered. This means that a set of constants $\{M^k_{i,j,s}\}$ has to be determined. Let $(i,j) \in \mathcal{C}$ and $t_k \in \mathcal{T}^-$. $M^k_{i,j,s}$ and $M^{k+1}_{i,j,s}$ need to be chosen so that any pair of reachable relative positions $\mathbf{p}^k_{i,j}$ and $\mathbf{p}^{k+1}_{i,j}$ satisfy separation constraints when $\delta^k_{i,j,s} = 0$, *i.e.*

$$\langle \mathbf{p}^k_{i,j} | \mathbf{e}(\theta_s) \rangle \geq \overline{D} - M^k_{i,j,s}$$

$$\langle \mathbf{p}^{k+1}_{i,j} | \mathbf{e}(\theta_s) \rangle \geq \overline{D} - M^{k+1}_{i,j,s},$$

where $\overline{D}$ is an upper bound of $D_{i,j}^k$. The value of $M_{i,j,s}^k$ leading to the tightest formulation of $\mathcal{P}_{MIL}$ is thus given by

$$M_{i,j,s}^k = \max_{\mathbf{p}_{i,j}^k} \{\overline{D} - \langle \mathbf{p}_{i,j}^k | \mathbf{e}(\theta_s) \rangle\} \tag{25}$$

Alonson-Ayuzo *et al.* gave a tight MILP formulation of conflict-free motion planning with speed and altitude manœuvres [12]. In [12], 2D paths are fixed and only velocity or flight level vary. It is then straightforward to calculate $\{M_{i,j,s}^k\}$ so that (25) is verified. If aircraft are allowed to change their headings, finding $\{M_{i,j,s}^k\}$ involves the resolution of a maximisation problem for each $(i,j) \in \mathcal{C}, t_k \in \mathcal{T}$ and $s \in \mathcal{S}$, which may be long and inefficient. Instead of applying such a process, it is assumed that, in optimal trajectories, maximum deviations from reference trajectories do not exceed 10% of the total reference trajectories lengths. This assumption is reasonable considering the dynamics of aircraft and the constraints that they have to be on their reference trajectories at the initial and final times. It was also verified with a comfortable margin on the experimental tests described in Section VI.

Let $\mathbf{p}_i^{ref}(t), t \in [0,T]$, be the reference trajectory of i and $\mathbf{d}_i(t), t \in [0,T]$, the deviation from this trajectory, *i.e.* $\mathbf{p}_i(t) = \mathbf{p}_i^{ref}(t) + \mathbf{d}_i(t), \forall t \in [0,T]$. If d_i^{ref} and d_j^{ref} are the maximum deviations of i and j implied by the assumption stated above, it is straightforward that, $\forall t_k \in \mathcal{T}, \forall s \in \mathcal{S}$,

$$\langle \mathbf{p}_{i,j}^k | \mathbf{e}(\theta_s) \rangle \leq \langle \mathbf{p}_{i,j}^{ref}(t_k) | \mathbf{e}(\theta_s) \rangle + d_i^{ref} + d_j^{ref} \tag{26}$$

$$\langle \mathbf{p}_{i,j}^k | \mathbf{e}(\theta_s) \rangle \geq \langle \mathbf{p}_{i,j}^{ref}(t_k) | \mathbf{e}(\theta_s) \rangle - d_i^{ref} - d_j^{ref} \tag{27}$$

From (27), it comes that the solution of (25) is

$$M_{i,j,s}^k = \overline{D} - \langle \mathbf{p}_{i,j}^{ref}(t_k) | \mathbf{e}(\theta_s) \rangle + d_i^{ref} + d_j^{ref} \tag{28}$$

Moreover, (26) implies that if there exists $s \in \mathcal{S}$ such that, for some $t_k \in \mathcal{T}$,

$$\langle \mathbf{p}_{i,j}^{ref}(t_k) | \mathbf{e}(\theta_s) \rangle + d_i^{ref} + d_j^{ref} < \overline{D}$$

then the separation constraint associated with (i,j), t_k and s may be dropped as it will not be satisfied. Similarly, (26) implies that if, for any $t_k \in \mathcal{T}$, there exists $s \in \mathcal{S}$:

$$\langle \mathbf{p}_{i,j}^{ref}(t_k) | \mathbf{e}(\theta_s) \rangle - d_i^{ref} - d_j^{ref} \geq \overline{D}$$

then all separation constraints for conflict (i,j) and time t_k may also be dropped as separation is guaranteed to be satisfied. Through such considerations, the number of constraints and binary variables is decreased for most problems and a much tighter formulation is obtained.

A similar approach could be applied to tighten constraints (19)-(20). Finally, the best value for the constant M' that appears in constraint (22) is the sum of the lower and upper bounds on velocity.

B. Valid inequalities

It is also possible to tighten the model by appending redundant constraints, hence reducing the feasible space while keeping optimal solutions. Several improvements were added to the model by considering specific properties of the application; two such simplifications are detailed below.

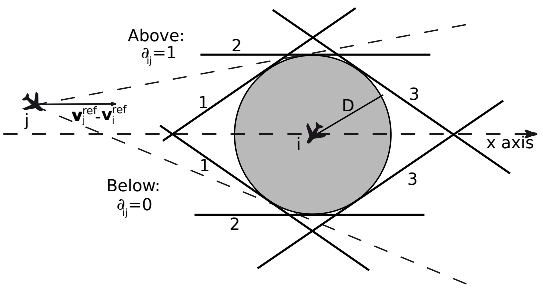


Fig. 4. Representation of valid inequalities

Let i and j be two aircraft in potential conflict. A rotation is applied to the moving frame $\mathbf{R}_i$ so that relative speed on the reference trajectories $\mathbf{v}_{i,j}^{ref}$ is collinear to the x-axis and points in the same direction, as in Figure 5. Experimental tests conducted on pairwise conflicts showed that when i and j cross with an angle larger than 30°, optimal solutions never plan that j performs a u-turn in the frame $\mathbf{R}_i$. Let $\mathcal{C}_{30^\circ}$ be the set of such potential conflicts. Assuming that this "no u-turn" hypothesis is valid, two sets of redundant constraints may be added to the problem. It should be observed that if the assumption is not verified, the inequalities below might prevent the solver from finding the optimal solution, but as long as the problem remains feasible the solution is a good candidate to initialise the NLP.

It is first important to state directly in the model that, under this assumption, $\forall (i,j) \in \mathcal{C}_{30^\circ}$, j passes either below or above $\mathbf{C}_i$. This statement avoids the exploration of many useless combinations when setting the values of $\{\delta_{i,j,s}^k\}$. The expression of this redundant constraint involves one new binary variable $\delta_{i,j}$ for each potential conflict, such that $\delta_{i,j} = 1$ when j goes above $\mathbf{C}_i$ and $\delta_{i,j} = 0$ when it goes below (see Figure 4). Let $\mathcal{S}^+$ be the set of tangents above the circle and $\mathcal{S}^-$ the set of tangents below the circle, the valid inequalities are formulated: $\forall (i,j) \in \mathcal{C}_{30^\circ}, \forall t_k \in \mathcal{T}$,

$$\sum_{s \in \mathcal{S}^+} \delta_{i,j,s}^k \leq \delta_{i,j}, \forall k \tag{28a}$$

$$\sum_{s \in \mathcal{S}^-} \delta_{i,j,s}^k \leq 1 - \delta_{i,j}, \forall k \tag{28b}$$

Under the same assumption, there also has to be a temporal progression in the choice of the separating tangents from left to right. $\mathcal{S}^+$ and $\mathcal{S}^-$ are then ordered so that $s < s' \in \mathcal{S}^+$ (or $s < s' \in \mathcal{S}^-$) implies that the binary variable associated with tangent s is set to 1 before the binary variable associated with s'. An ordering is represented by the numbering of tangents above and below the circle in Figure 4. These constraints are then formulated as: $\forall (i,j) \in \mathcal{C}_{30^\circ}, \forall t_k \in \mathcal{T}$,

$$\sum_{s \in \mathcal{S}^+} s\delta_{i,j,s}^k \leq \sum_{s \in \mathcal{S}^+} s\delta_{i,j,s}^{k+1} \tag{28c}$$

$$\sum_{s \in \mathcal{S}^-} s\delta_{i,j,s}^k \leq \sum_{s \in \mathcal{S}^-} s\delta_{i,j,s}^{k+1} \tag{28d}$$

C. Relaxation of the separation constraints

When considering a large number of aircraft in potential conflict, runtime may be far over the operational requirements.

It would indeed be exaggerated to spend more than one or two minutes looking for a resolution to conflicts that are to happen in five to ten minutes. A way of tackling this issue is to add a runtime limit of say 60 seconds when solving the MILP and use the best feasible solution found as a candidate starting point for the optimisation of $\mathcal{P}_{NL}$. It is however not even sure that a feasible solution is found within the 60 seconds, which is why separation constraints are relaxed by adding non negative slack variables penalised in the criterion. Let $\{sl_{i,j}^k\}$ be the slack variables; once relaxed, (15) and (16) become: $\forall(i,j) \in \mathcal{C}, \forall t_k \in \mathcal{T}^-$

$$\langle \mathbf{p}_{i,j}^k | \mathbf{e}(\theta_s) \rangle \geq D_{i,j}^k - M(1-\delta_{i,j,s}^k) - sl_{i,j}^k, \forall s \in \mathcal{S} \quad (29)$$

$$\langle \mathbf{p}_{i,j}^{k+1} | \mathbf{e}(\theta_s) \rangle \geq D_{i,j}^k - M(1-\delta_{i,j,s}^k) - sl_{i,j}^k, \forall s \in \mathcal{S} \quad (30)$$

$$sl_{i,j}^k \geq 0 \quad (31)$$

and the term $\sum_{(i,j)\in\mathcal{C},t_k\in\mathcal{T}^-} \mu \times sl_{i,j}^k$ is added to the criterion with $\mu >> \overline{U}$. Once separation constraints are relaxed, it is much easier to get a solution that is at least feasible with regard to every constraint but the separation constraints. By penalising the slack variables, the minimisation searches solutions in which slack variables are equal to 0, which implies that the separation constraints are satisfied.

VI. Computational experiments

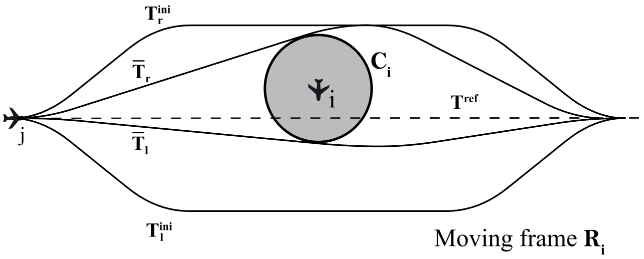
A. Sensitivity to initialisation

When solving a nonconvex NLP, classical iterative algorithms only converge to a local minimum. This solution and the speed of convergence both depend on the point from which the algorithm is started. The sensitivity of the solution of $\mathcal{P}_{NL}$ to initialisation is first illustrated on a simple case before studying more complex scenarios.

A pairwise potential conflict between two aircraft i and j is considered and it is assumed that all other aircraft in the vicinity are far enough to be disregarded. The problem was solved with two different initial solutions $\mathbf{T}_r^{ini}$ and $\mathbf{T}_l^{ini}$. On trajectories $\mathbf{T}_r^{ini}$ (resp. $\mathbf{T}_l^{ini}$) both aircraft make a 20° turn to the right (resp. on the left) for 2 minutes and then go straight until they make another 20° turn to recover their reference trajectories. Once represented in the moving frame $\mathbf{R}_i$, if j follows $\mathbf{T}_r^{ini}$ (resp. $\mathbf{T}_l^{ini}$) it passes above (resp. below) $\mathbf{C}_i$.

$\mathcal{P}_{NL}$ was solved with SNOPT[1], a software package for solving large-scale optimisation problems. It uses Sequential Quadratic Programming (SQP), an iterative procedure which approximates the NLP by a quadratic programming subproblem at each step. More details on SQP may be found in [27].

Solutions $\overline{\mathbf{T}}_r$ and $\overline{\mathbf{T}}_l$ are represented next to their matching initial solutions in Figure 5. Solution $\overline{\mathbf{T}}_r$ was found to be 150% more expensive than $\overline{\mathbf{T}}_l$. Resolution kept the trajectory on the same side of $\mathbf{C}_i$ as in the initial solution, hence leading to a clearly suboptimal solution $\overline{\mathbf{T}}_r$.



$\mathbf{T}_r^{ini}$
$\overline{\mathbf{T}}_r$
$\mathbf{C}_i$
i
$\mathbf{T}^{ref}$
j
$\overline{\mathbf{T}}_l$
$\mathbf{T}_l^{ini}$
Moving frame $\mathbf{R}_i$

Fig. 5. Pairwise conflict with two sets of resolutions

B. Generation of data sets

In order to illustrate the performances of the algorithms presented in this article, three generic data sets were chosen for their complexities and the diversity of the difficulties they offer. They focus on clusters of aircraft in potential conflict and they were designed in order to cover situations ranging from what may currently be handled by ATCOs to much more complex conflicts. The aircraft of a data set are entirely described by their initial states (positions and speeds), their final goal positions and the bounds on their speeds and accelerations. The environment is given by the equations of the edges of the polygonal segregated areas. For simplicity, all aircraft have identical performances. Parameters were chosen to represent generic performances found in the BADA database[2]: nominal velocity $V_{nom} = 500$ kts, $\overline{V} = 1.05 \times V_{nom}$, and $\underline{V} = 0.92 \times V_{nom}$. The time interval is 10 minutes long and is split in 10 sampling periods (i.e. $K = 10$). The number of sampling periods was set on the basis of a previous work [28] in which the effect of the number of periods was studied thoroughly on a reduced number of data sets. Although the model was not as complete as the one described in this article, the global behaviour of the algorithm was close enough to follow the conclusions that were drawn in [28] on this topic.

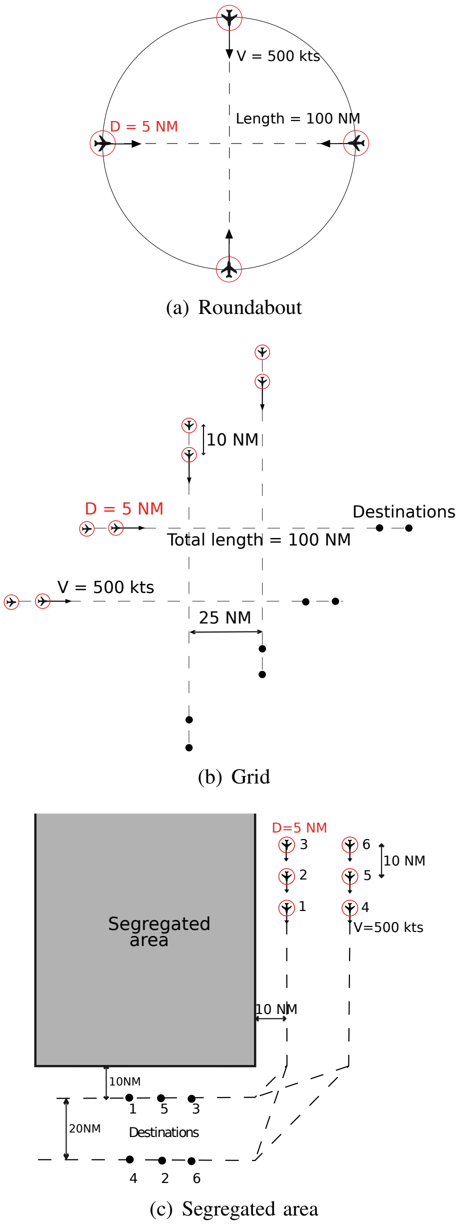
The first type of data sets is a classical academic example in automated ATC [7], [29]. Several aircraft are initially set on a 50 NM radius circle and must reach the opposite extremity of the diameter by the end of the time range. Figure 6(a) gives a representation of the initial situation with four aircraft. Its difficulty lies in the strong dependency between all the conflicts. It is not possible to try to solve one conflict without impacting the others. The optimal solution is known for aircraft with equal performances: all aircraft take a turn in the same direction, as if in a roundabout.

Another configuration that was considered interesting involves parallel aircraft flows intersecting perpendicularly (see Figure 6(b)). It aims at studying the behaviour of the automated ATC when several consecutive conflicts have to be handled in a very structured environment. It also illustrates a specificity of air traffic because controlled airspace is currently organized in routes that aircraft have to follow.

The third type of data sets illustrates the ability of avoiding a segregated area while managing a complex situation. The aircraft are initially organised in two parallel flows and they have to reach final points to which the shortest path infringes on the segregated area. In order to make the situation more complex, the flows are mixed: aircraft belonging to the same

[1] website : http://www.sbsi-sol-optimize.com/asp/sol_product_snopt.htm

[2] BADA is the Base of Aircraft Data of EUROCONTROL. It is available online at http://www.eurocontrol.int/eec/public/standard_page/proj_BADA.html



(a) Roundabout

(b) Grid

(c) Segregated area

Fig. 6. Representations of the benchmark data sets

initial flow do not all aim at the same final route. Figure 6(c) gives a possible implementation of this configuration with three aircraft per flow.

The three generic situations were used to generate a complete benchmark. For each scenario described above, four configurations were chosen with increasing numbers of aircraft and conflicts. Their properties are summarised in Table I. n_v, n_b, and n_c are the numbers of variables, binary variables and constraints in the associated tightened MILP, respectively. The high symmetry in these configurations was partly broken by adding random terms in the initial positions of the aircraft. Each aircraft was moved on its reference trajectory according to a distance that was randomly sampled from a uniform distribution on $[-3\text{ NM}, 3\text{ NM}]$. The tests were conducted on 100 randomly generated data sets for each configuration in Table I.

TABLE I
SUMMARY OF THE DATA SETS

data	scenario	$\lvert\mathcal{A}\rvert$	$\lvert\mathcal{C}\rvert$	n_v	n_b	n_c
G-01	Grid	6	15	732	125	3404
G-02	Grid	6	15	774	153	3516
G-03	Grid	8	28	1142	268	4972
G-04	Grid	12	66	1866	476	7838
R-01	Roundabout	3	3	342	51	1632
R-02	Roundabout	4	6	512	102	2308
R-03	Roundabout	5	10	746	194	3146
R-04	Roundabout	6	15	1008	303	4050
S-01	Seg. Area	4	6	928	414	3444
S-02	Seg. Area	5	10	1370	664	4860
S-03	Seg. Area	6	15	1896	977	6498
S-04	Seg. Area	7	21	2506	1346	8358

C. Computational results

This section reports the results obtained when solving $\mathcal{P}_{NL}$ for the data sets described in the preceding section. As it was observed in Section VI-A that solving $\mathcal{P}_{NL}$ was sensitive to initialisation, three options were considered in order to provide a good starting point. Such a resolution is often referred to as a *warm start*.

Solving the complete MILP, presented in Section IV, provides the optimal solution of an approximate problem. It ensures that the solver process starts from a good point that might even be in the neighbourhood of the NLP optimum in some cases. The MILP solution being feasible for $\mathcal{P}_{NL}$, the current best solution of the solver is always feasible, which means that it may be stopped at any moment to return an efficient motion planning for the aircraft. This *any time* property is very useful in an application such as ATC where reactivity and robustness are important. The hybrid warm start optimisation of $\mathcal{P}_{NL}$ from the solution of $\mathcal{P}_{MIL}$ is noted $\mathcal{S}_h$. Due to the fact that waiting for the MILP to converge may take too much time, another candidate starting point was obtained by adding a time limit of 60 seconds when solving $\mathcal{P}_{MIL}$. The separation constraints were relaxed, as described in V-C, to make sure that a solution is always found. This alternative process is noted $\mathcal{S}_h^{60}$. As they are already available without any further calculation, reference trajectories were provided as a third candidate starting point to $\mathcal{P}_{NL}$ resolution. The reference trajectories satisfy every constraint but separation constraints. In the simple pairwise conflict studied in Section VI-A, this starting point also leads to the best solution. Warm start optimisation from reference trajectories is noted $\mathcal{S}_{ref}$.

When solving $\mathcal{P}_{MIL}$, the lower bounds on velocity (22)-(24) were withdrawn because the recovery of the planned positions and speeds associated with the minimisation of acceleration are more likely to lead to velocities above nominal speed. A post-process routine was implemented in order to verify that lower bounds are satisfied and solve $\mathcal{P}_{MIL}$ again with the constraints (22)-(24) if it is not the case. However this second resolution was never needed because no violation of (22)-(24) was detected by the routine during the tests.

The values of the parameters N and Q involved in the linearisation of the quadratic constraints on velocity and acceleration did not impact computation time seriously, which is why they were set to rather high values: $N = 40$ and $2Q = 10$.

TABLE II
COMPUTATIONAL RESULTS FOR EACH ALGORITHM

data	$\%_0$	$\%_r$	$\%_h$	$\%_h^{60}$	z_0	z_r	z_M	z_h	z_M^{60}	z_h^{60}
G-01	100	100	100	100	19.3	18.3	26.2	12.9	26.2	12.9
G-02	91	100	100	100	42.4	16.3	17.2	14.3	17.2	14.3
G-03	100	100	100	100	42.1	36.4	25.0	20.2	25.1	20.2
G-04	88	99	100	99	88.3	53.1	35.4	27.4	47.7	40.8
R-01	100	100	100	100	11.6	9.8	9.9	8.9	9.9	8.9
R-02	100	100	100	100	17.5	15.6	16.3	14.1	16.3	14.1
R-03	87	88	100	100	40.6	28.1	24.9	21.5	24.9	21.5
R-04	62	89	100	100	58.8	34.6	34.9	29.6	35.4	29.8
S-01	99	99	100	100	48.9	48.4	49.0	48.4	49.0	48.4
S-02	99	99	100	100	65.4	60.5	61.2	60.5	61.2	60.5
S-03	99	97	100	100	76.5	72.6	73.7	72.6	73.7	72.6
S-04	100	99	100	100	96.3	84.9	85.9	84.8	86.1	85.0

TABLE III
COMPUTATION TIME FOR EACH ALGORITHM (SECONDS)

data	t_0	t_r	t_{NL}	t_M	t_h	t_{NL}^{60}	t_M^{60}	t_h^{60}
G-01	8	3	3	6	8	3	6	8
G-02	5	2	3	3	6	3	3	6
G-03	10	6	4	33	37	4	26	30
G-04	61	25	11	658	670	14	60	74
R-01	1	1	1	1	2	1	1	2
R-02	2	2	1	2	3	1	2	3
R-03	3	2	2	23	24	2	23	24
R-04	6	4	2	98	100	2	59	61
S-01	4	3	2	3	5	2	3	5
S-02	8	4	4	6	10	4	6	10
S-03	18	8	6	17	23	6	17	23
S-04	29	11	9	54	63	11	41	52

On the other hand, computation time is very sensitive to the number of separation tangents, which is why $|\mathcal{S}|$ was set to its minimal value, i.e. $S = 4$.

$\mathcal{P}_{MIL}$ was solved by running the commercial solver Gurobi[3]. As most MILP solvers, Gurobi explores the search tree with a branch and bound method and tightens the continuous relaxation by generating valid inequalities. The reader is referred to [30] for an overview of the current MILP solvers and the advanced techniques they employ. The implementation of the improvements described in Sections V-A and V-B sped up the resolution of $\mathcal{P}_{MIL}$ by a factor of two to ten. Most of this acceleration is due to valid inequalities. As in Section VI-A, $\mathcal{P}_{NL}$ was solved with SNOPT. The default options of both solvers were used. Computations were performed on a PC Intel Xeon 2.80 Ghz and 12 Gbytes of RAM.

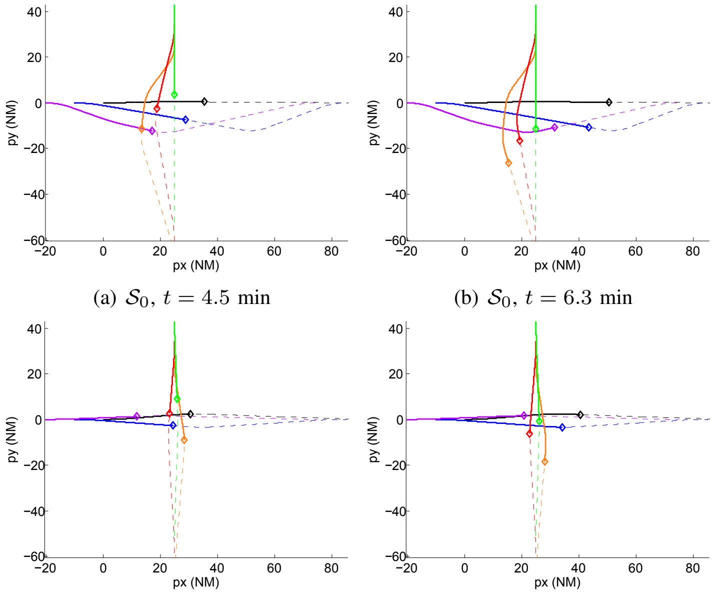
Tables II and III report comparative results for the three warm starts introduced above and for a resolution where no starting point is provided to the solver, which is noted $\mathcal{S}_0$. In Table II, column headers $\%_0$, $\%_r$, $\%_h$ and $\%_h^{60}$ are the percentages of data sets for which a feasible solution was found by $\mathcal{S}_0$, $\mathcal{S}_{ref}$, $\mathcal{S}_h$ and $\mathcal{S}_h^{60}$, respectively. As finding a feasible solution is the primary objective, the other columns give average results over the data sets for which every method found a feasible solution. z_0, z_r, z_h and z_h^{60} are the costs of the best solutions found by $\mathcal{S}_0$, $\mathcal{S}_{ref}$, $\mathcal{S}_h$ and $\mathcal{S}_h^{60}$. The costs z_M and z_M^{60} of the initial solutions found by solving $\mathcal{P}_{MIL}$ with no time limit and with a 60 seconds time limit were added. Through the construction of $\mathcal{P}_{MIL}$ they provide an upper bound of z_h and z_h^{60} and indicate how much the resolution of $\mathcal{P}_{NL}$ was able to improve this starting point. In Table III, t_0, t_r, t_h, t_h^{60} are the elapsed times (seconds) to find these solutions. Computation times of the hybrid algorithm, t_h and t_h^{60}, are decomposed to give insight on the proportion of time spent in the resolutions of $\mathcal{P}_{MIL}$ (t_M and t_M^{60}) and $\mathcal{P}_{NL}$ (t_{NL} and t_{NL}^{60}).

A first analysis of the results in Table II focuses on feasibility. The SQP algorithm implemented in SNOPT alternates between a phase where a quadratic criterion subject to a linear approximation of the constraints is solved to get the next iterate which might not satisfy all nonlinear constraints, and an elastic phase during which feasibility is recovered with respect to a given tolerance. In most cases, an infeasible solution satisfies the linear constraint but does not meet the feasibility tolerance with respect to nonlinear constraints, although SNOPT minimised these nonlinear infeasibilities. This was the case when $\mathcal{S}_0$ or $\mathcal{S}_{ref}$ did not converge to a feasible solution. The trajectories characterised by infeasible solutions thus look implementable but slightly violate either the upper bound on velocity, the separation constraints, or both. Contrary to $\mathcal{S}_0$ and $\mathcal{S}_{ref}$, the hybrid algorithm $\mathcal{S}_h$ was always able to find a feasible solution because the global search realised when solving $\mathcal{P}_{MIL}$ was successful in providing a starting point that is feasible for $\mathcal{P}_{NL}$. In this respect, it is also remarkable that $\mathcal{S}_{ref}$ always performed better than $\mathcal{S}_0$. This shows that even a partially feasible solution constitutes a valuable starting point.

By focusing on the results of $\mathcal{S}_0$, it appears that it was more difficult to find a feasible solution for G-02 than for G-03 although G-03 involves more aircraft. This behaviour is explained by the fact that G-02 simulates the encounter of two trails of three aircraft while G-03 involves trails of two aircraft (G-03 is the scenario drawn in Figure 6(b)). In Figure 7, an infeasible solution produced by $\mathcal{S}_0$ for G-02 is compared with the feasible solution produced by $\mathcal{S}_h$. The diamond symbols give the positions of the aircraft at t, and dotted lines correspond to the remaining parts of the trajectories. In the solution of $\mathcal{S}_0$, the whole vertical trail goes behind the horizontal trail, which results in a 2.5 NM minimum distance between the first aircraft of the vertical trail and the last aircraft of the horizontal trail. Although the solution of $\mathcal{S}_h$ is feasible and the associated trajectories look implementable, it results in a situation that ATCOs may still consider too dangerous because the aircraft of the vertical trail go between successive aircraft of the horizontal trail. In order to find more robust trajectories, the uncertainties should be explicitly included in the models. On the other hand, it is interesting to observe in Figure 8 that trajectories found by $\mathcal{S}_h$ are naturally structured as in a roundabout while the more costly solution found by $\mathcal{S}_{ref}$ does not present any specific structure.

Solution costs also followed the same hierarchy between the models. Solutions found by $\mathcal{S}_h$ were on average 15% better than $\mathcal{S}_{ref}$ solutions and 32% better than $\mathcal{S}_0$. The solutions of $\mathcal{P}_{MIL}$ were already cheaper than those of $\mathcal{S}_{ref}$ and $\mathcal{S}_0$, and the resolution of $\mathcal{P}_{NL}$ improved them by an average 10%. $\mathcal{S}_{ref}$

[3] Gurobi website: http://gurobi.com



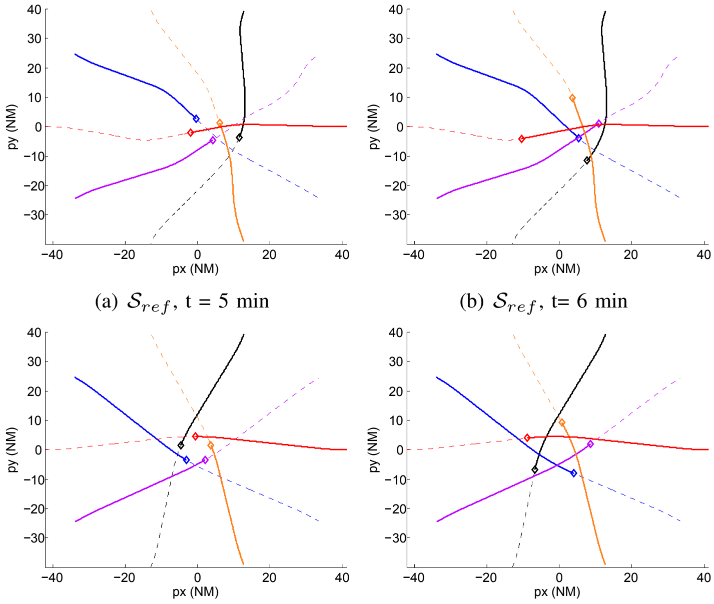
(a) $\mathcal{S}_0$, $t = 4.5$ min

(b) $\mathcal{S}_0$, $t = 6.3$ min

(c) $\mathcal{S}_h$, $t = 4.0$ min

(d) $\mathcal{S}_h$, $t = 5.2$ min

Fig. 7. Infeasible (above) and feasible trajectories (below) for G-02; a specific color is assigned to each aircraft



(a) $\mathcal{S}_{ref}$, t = 5 min

(b) $\mathcal{S}_{ref}$, t= 6 min

(c) $\mathcal{S}_h$, t = 5 min

(d) $\mathcal{S}_h$, t = 6 min

Fig. 8. Infeasible (above) and feasible trajectories (below) for R-03; a specific color is assigned to each aircraft

also proved to be an efficient alternative to $\mathcal{S}_{MIL}$ in terms of cost.

Finally, the runtime of $\mathcal{S}_h$ was bigger in every case although t_{NL} and t_{NL}^{60} are slightly smaller than t_0 and t_r. The starting point obtained by solving $\mathcal{P}_{MIL}$ accelerates the convergence of the resolution of $\mathcal{P}_{NL}$ but $\mathcal{P}_{MIL}$ is a large scale MILP which may be time-consuming. Nevertheless, computational effort remains below 40 seconds for most data sets. Computation time would only be unacceptable for the biggest "grid" and "roundabout" data sets, G-04 and R-04, and could be found excessive for some S-04 data sets. The difficulties encountered when solving a "roundabout" scenario are mostly due to its highly symmetric structure which is usually a major obstacle to combinatorial optimisation. The last "grid" data set, G-04, was longer to solve because it involves many more aircraft. It is however important to notice that such complex situations are never met in the current air traffic. Situations where six interacting aircraft are in potential conflict are exceptional and the airspace structure forbids such symmetric trajectories as in "grid" and "roundabout" scenarios. If such a situation is encountered in real traffic, the quick resolution obtained by adding a time limit of 60 seconds proved to be efficient as it only failed to find a feasible solution once and still provides solutions whose costs are smaller than z_0 and z_r, even if a 50 % cost increase was observed for G-04.

VII. Conclusion

The objective of the approach presented in this article is to solve the problem of conflict-free motion planning for multiple aircraft on the same flight level with trajectory recovery. One of the major advancements in this paper is a consistent approximation scheme from the natural continuous-time representation of the problem to a discrete-time linear approximation. In particular, it connects formulations that are usually treated independently in the literature. Moreover, the approximations are chosen in order to satisfy the requirements for separation between aircraft at all times with accurate constraints. The operational concept of 4D contract is also taken into account by explicitly including in the model the recovery of reference trajectories after the necessary manœuvres. The second important contribution is a hybrid algorithm in which a warm start optimisation of a nonlinear model is performed from the solution of a linear approximation. This algorithm enables taking into consideration the real geometry of nonlinear constraints while guaranteeing to find a feasible solution when the linear model has one. Experimental tests were conducted on a large benchmark including three air traffic scenarios, representative of several difficulties that may be encountered. These tests highlighted the positive impact of the hybridisation in terms of number of feasible solutions and of cost of these solutions. It was however observed that the resolution of the mixed integer linear model could take too much time for an operational implementation. A significant improvement in runtime with only a minor deterioration of the solutions was achieved by stopping this resolution after one minute when the solver had not converged yet.

On a theoretical level, a perspective that was opened by this work deals with the development of an algorithm aiming at global optimality of the NLP by iteratively improving bounds on its criterion. Indeed, it is possible to build a linear approximation whose minimum cost is a lower bound of the optimal cost of the NLP. This approximation may be obtained by a similar process as the one described in this article except that, when linearising quadratic constraints, chords should be replaced with tangents and, reciprocally, tangents with chords.

Our current research on the subject focus on improving the implementability of the models with the aim of integrating it in a decision support system. In order to meet the constraints imposed by decision makers, additional research has to be conducted on four levels. The model has some limitations, such as the fact that constraints on longitudinal and lateral accelerations are not decoupled, and that the criterion does not explicitly formulate the operational cost of trajectories. More precise constraints on acceleration and a cost function based on fuel consumption are to be included in the model. Moreover, as the definition of an ideal criterion is still an open issue, other criteria such as deviation from the reference trajectory could be investigated. In order to extend the applicability of this automated conflict resolution to aircraft whose altitude is not stabilised, a 3D model is under study. Uncertainties should also be explicitly added in the problem in order to obtain robust conflict-free motion planning. It must however be noted that these last two improvements give rise to much more complex problems, that may only be tractable for small instances unless some simplifying assumptions are made. Finally, the benchmark proposed here was useful to demonstrate the strengths and limits of this algorithm but it remains too artificial. There would be a need to test this method on operational data. Full scale real-life data sets were already collected and still have to be used as other entries of our benchmark. After addressing these issues, the solver could be implemented as a module activated either periodically or upon request of the controllers. It will then be possible to assess the impact of the system in an operational context.

References

[1] "SESAR definition phase, deliverable 3: The ATM target concept," SESAR Joint Undertaking, Tech. Rep., 2007.

[2] J. Hopcroft, J. Schwartz, and M. Shahir, "On the complexity of motion planning for multiple independent objects: PSPACE-hardness of the warehouseman's problem," *International Journal of Robotics Research*, vol. 3, pp. 76–88, 1984.

[3] C. R. Hargraves and S. W. Paris, "Direct trajectory optimization using nonlinear programming and collocation," *AIAA Journal of Guidance, Control, and Dynamics*, vol. 10, pp. 3–12, 1987.

[4] J. K. Kuchar and L. C. Yang, "A review of conflict detection and resolution modeling methods," *IEEE Transactions on Intelligent Transportation Systems*, vol. 1, pp. 179–189, 2000.

[5] B. Kirwan and M. Flynn, "Identification of air traffic controller conflict resolution strategies for the CORA project," in *ATM Seminar*, 2001.

[6] T. Farley and H. Erzberger, "Fast-time simulation evaluation of a conflict resolution algorithm under high air traffic demand," in *ATM Seminar*, 2007.

[7] N. Durand, J.-M. Alliot, and J. Noailles, "Automatic aircraft conflict resolution using genetic algorithms," in *Symposium on Applied Computing*, 1996.

[8] A. Bicchi and L. Pallotino, "Optimal cooperative conflict resolution for air traffic management systems," *IEEE Transactions on Intelligent Transportation Systems*, pp. 100–109, 2000.

[9] L. Pallottino, E. Feron, and A. Bicchi, "Conflict resolution problems for air traffic management systems solved with mixed integer programming," *IEEE Transactions on Intelligent Transportation Systems*, vol. 3, no. 1, pp. 3 –11, 2002.

[10] A. Vela, S. Solak, J. Clarke, W. Singhose, E. Barnes, and E. Johnson, "Near real-time fuel-optimal en route conflict resolution," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 4, pp. 826 –837, 2010.

[11] M. Christodoulou and S. Kodaxakis, "Automatic commercial aircraft-collision avoidance in free flight: the three-dimensional problem," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 2, pp. 242 – 249, 2006.

[12] A. Alonso-Ayuso, L. Escudero, P. Olaso, and C. Pizarro, "Conflict avoidance: 0-1 linear models for conflict detection & resolution," *TOP*, pp. 1–20, 2011.

[13] A. Richards and J. How, "Aircraft trajectory planning with collision avoidance using mixed integer linear programming," in *American Control Conference*, vol. 3, 2002, pp. 1936–1941.

[14] P. K. Menon, G. D. Sweriduk, and B. Sridhar, "Optimal strategies for free flight air traffic conflict resolution," *Journal of Guidance, Control, and Dynamics*, vol. 22, pp. 202–211, 1997.

[15] P. Belotti, J. Lee, L. Liberti, F. Margot, and A. Wächter, "Branching and bounds tightening techniques for non-convex MINLP," *Optimization Methods and Software*, vol. 24, no. 4-5, pp. 597–634, 2009.

[16] S. Guibert and L. Guichard, "4D trajectory management through contract of objectives," in 9^{th} *Innovative Research Workshop & Exhibition*, 2010.

[17] J. T. Betts, *Practical methods for optimal control using nonlinear programming*, ser. Advances in Design and Control, S. for Industrial and applied Mathematics (SIAM), Eds., 2001.

[18] R. A. Paielli, "Modeling maneuver dynamics in air traffic conflict resolution," *Journal of Guidance, Control, and Dynamics*, vol. 26, no. 3, pp. 407–415, 2003.

[19] C. G. E. Boender, A. H. G. Rinnooy Kan, L. Stougie, and G. T. Timmer, "Global optimization: a stochastic approach," in *Numerical techniques for stochastic systems*, 1980, pp. 387–394.

[20] G. Dantzig, "The general simplex method for minimizing a linear form under inequality constraints," *Pacific Journal of Mathematics*, vol. 5, no. 2, pp. 183–195, 1955.

[21] Y. Nesterov and A. Nemirovski, *Interior point polynomial methods in convex programming*, S. S. in Applied Mathematics, Ed., 1994.

[22] W. Niedringhaus, "Stream Option Manager (SOM): Automated integration of aircraft separation, merging, stream management, and other air traffic control functions," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 25, pp. 1269–1280, 1995.

[23] J. Omer and T. Chaboud, "Tactical and post-tactical air traffic control methods," in 9^{th} *Innovative Research Workshop & Exhibition*, 2010, pp. 11–18.

[24] S. J. Guy, M. C. Lin, and D. Manocha, "Modeling collision avoidance behavior for virtual humans," in 9^{th} *International Conference on Autonomous Agents and Multiagent Systems*, 2010, pp. 575–582.

[25] J. Peng and S. Akella, "Coordinating multiple robots with kinodynamic constraints along specified paths," *The International Journal of Robotics Research*, vol. 24, pp. 295–310, 2005.

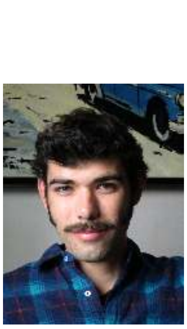
[26] R. Gomory, "Outline of an algorithm for integer solutions to linear programs," *Bulletin of the American Mathmatical Society*, vol. 64, pp. 275–278, 1958.

[27] J. Nocedal and S. Wright, *Numerical optimization*, 2^{nd} edition ed., ser. Springer series in operations research. Springer, 2006.

[28] J. Omer and J.-L. Farges, "Automating air traffic control through nonlinear programming," in *International Conference on Research in Air Transportation (ICRAT)*, http://www.icrat.org, (on line), 2012.

[29] C. Peyronne, D. Delahaye, and L. Lapasset, "Conflict-free trajectory optimization using b-splines and genetic algorithm," in 9^{th} *Innovative Research Workshop & Exhibition*, 2010, pp. 213–218.

[30] A. Lodi and J. T. Linderoth, "MILP software," *Encyclopedia for Operations Research and Management Science, Wiley*, 2011.



Jérémy Omer was born in Paris, France, in 1982. He received a B.S. degree in electric engineering from Supelec, France, in 2006 and received a M.S. degree in applied mathematics from Polytechnique Montreal, Canada, in a double graduation program in 2006. He received the Ph.D degree in 2013 from ISAE, Toulouse, for research conducted under the guidance of Dr. G. Verfaillie and Dr T. Chaboud at Onera, the French Aerospace Lab. His research interests include optimisation techniques applied to logistics, robotics, and air-traffic systems.

Jean-Loup Farges was born in Clamart, France, in 1956. In 1978 he received a degree in electric engineering from ENSEEIHT, Toulouse, France, and a M.S. degree in automatic control from Paul Sabatier University in Toulouse. He received his Ph.D degree from ISAE, Toulouse, in 1983. He has since been working as a Research Scientist for Onera, the French Aerospace Lab. Initially, his main area of interest was road traffic control. Since 2003 his research focuses on planning and execution control for aerial vehicles and on optimisation in air traffic control.